

## In the Claims

1. (Currently Amended) A method for testing at least one software application, the method comprising:

retrieving information descriptive of a state of operation of the software application being tested and a at least one graphics element rendered during deterministic execution of the software application being tested, wherein the information identifying identifies an executable feature associated with the at least one graphics element;

storing information related to an association between the executable feature and the at least one graphics element and the state of operation of the software application in a map data structure containing information related to at least one graphics element for testing, the association and information being stored in the map data structure during execution of the software application being tested;

selecting an executable feature from the map data structure based on the association stored in the map data structure[;]], wherein selecting the executable feature proceeds according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing;

executing the selected executable feature associated with the graphics element; and

dynamically updating the information related to the state of operation of the software application and the association in the map data structure upon execution of the executable feature.

1 2. (Currently Amended) The method of claim 1, further comprising: ~~[[,]]in~~  
2 response to ~~executing the executable feature~~:

3 dynamically updating information in the map data structure descriptive of  
4 at least one second graphics element resulting from the exposure of a new state of  
5 operation of the software application in response to the execution of the executable  
6 feature;

7 displaying a the at least one second graphics element;

8 retrieving information descriptive of the state of operation of the software  
9 application being tested and the a at least one second graphics element rendered  
10 during execution of the software being tested, the information including a second  
11 executable feature associated with the at least one second graphics element and the  
12 new state of operation of the software application;

13 storing the information descriptive of the new state of operation of the  
14 software application being tested and the a second executable feature in association  
15 with the at least one second graphics element in the map data structure during  
16 execution of the software being tested; and

17 executing the at least one second executable feature stored in association  
18 with the second graphics element.

19  
20 3. (Currently Amended) The method of claim 1 wherein the retrieving  
21 comprises capturing information pertaining to the graphics element and the state of  
22 operation of the software application being tested.

23  
24  
25 4. (Currently Amended) The method of claim 1, wherein the storing

1 includes updating an indicator associated with the at least one graphics element  
2 when the executable feature stored in association with the at least one graphics  
3 element is executed.

4  
5 5. **(Currently Amended)** The method of claim 1 wherein the storing  
6 includes organizing the retrieved information such that the executable feature  
7 stored in association with the at least one graphics element can be interpreted by  
8 a computer-executable application capable of accessing the retrieved  
9 information.

10  
11 6. **(Currently Amended)** The method of claim 1 wherein the storing  
12 includes organizing the retrieved information such that the executable feature  
13 stored in association with the at least one graphics element and the state of  
14 operation of the software application being tested can be interpreted by a user  
15 capable of accessing the retrieved information from memory.

16  
17 7. **(Canceled)**

18  
19 8. **(Currently Amended)** The method of claim ~~7~~ 1 wherein the selecting  
20 comprises selecting an executable feature not previously executed.

21  
22 9. **(Original)** The method of claim 8 wherein the selecting comprises reviewing  
23 an indicator to select an executable feature not previously executed.

1 10. **(Currently Amended)** The method of claim 7 1 wherein the selecting  
2 comprises selecting executable features in a depth-first mode of operation.

3  
4 11. **(Currently Amended)** The method of claim 7 1 wherein the selecting  
5 comprises selecting executable features in a breadth-first mode of operation.

6  
7 12. **(Canceled)**

8  
9 13. **(Currently Amended)** A system for generating a map, ~~that associates a~~  
10 ~~graphics element of a graphical user interface of a software application with an~~  
11 ~~executable feature of the software application~~ the system comprising:

12 a capture agent for retrieving information descriptive of a state of  
13 operation of a software application being tested and a plurality of graphics  
14 elements rendered during deterministic execution of the software application,  
15 the information including an executable feature associated with each graphics  
16 element;

17 an application driver for storing information in a map data structure  
18 related to an association between each executable feature and corresponding  
19 graphics element and a state of operation of the software application in a map  
20 data structure during execution of the software application being tested,  
21 wherein the map data structure contains information related to at least one  
22 graphics element for testing;

23 an application driver and for deterministically selecting one of the  
24 executable features stored in the map data structure based on the information  
25 stored in the map data structure, wherein deterministically selecting proceeds

1 according to a sequence determined by one of a plurality of deterministic modes  
2 for a systematic order of software application execution during testing; the  
3 association being stored in the map data structure during execution of the software  
4 application and;  
5 a command agent for executing the selected executable feature[[,]]; and  
6 ~~wherein the map data structure contains information related to at least one~~  
7 ~~graphics element for testing. an indicator for tracking a dynamic updating of the~~  
8 information related to the association and the state of operation of the software  
9 application in the map data structure upon the execution of the selected executable  
10 feature.

11  
12 14. **(Original)** The system of claim 13 wherein the capture agent is invoked by  
13 the application driver.

14  
15 15. **(Original)** The system of claim 13 wherein the capture agent submits the  
16 retrieved information to the application driver.

17  
18 16. **(Canceled)**

19  
20 17. **(Currently Amended)** The system of claim 13, wherein the application  
21 driver deterministically selects one of the executable features that has not been  
22 previously executed.

23  
24  
25 18. **(Currently Amended)** The system of claim ~~17~~ 13, wherein the application

1 driver reviews ~~an~~ the indicator to select the ~~one~~ executable feature.

2  
3 19. **(Currently Amended)** The system of claim 13, wherein the application  
4 driver deterministically selects executable features according to ~~in~~ a depth-first  
5 deterministic mode of operation.

6  
7 20. **(Currently Amended)** The system of claim 13 wherein the application  
8 driver deterministically selects executable features according to ~~in~~ a breadth-first  
9 deterministic mode of operation.

10  
11 21. **(Canceled)**

12  
13 22. **(Currently Amended)** A method for systematically invoking an executable  
14 feature of a software application having a graphical user interface, the method  
15 comprising:

16 retrieving information descriptive of a state of operation of a software  
17 application being tested and at least one graphics element rendered during  
18 deterministic execution of the software application, the information including an  
19 executable feature associated with the at least one graphics element;

20 storing information related to an association between the executable feature  
21 and corresponding graphics element and the state of operation of the software  
22 application in a map data structure to contain information related to at least one  
23 graphics element for testing, the association and information being stored in the  
24 map data structure during execution of the software application;

25 selecting from the map data structure at least one executable feature

1 associated with a graphics element that has not been previously executed, wherein  
2 selecting the at least one executable feature proceeds according to a sequence  
3 determined by one of a plurality of deterministic modes for a systematic order of  
4 software application execution during testing; and  
5 executing the selected at least one executable feature.

6  
7 23. **(Currently Amended)** The method of claim 22 further comprising, in  
8 response to executing the selected executable feature:

9 dynamically updating the information related to the association and the  
10 state of operation of the software application in the map data structure upon  
11 execution of the selected at least one executable feature;

12 displaying ~~another~~ a second graphics element[[:]]; ;

13 retrieving information descriptive of the ~~other~~ second graphics element  
14 rendered during execution of the software application and the state of operation of  
15 the software application being tested, the information including ~~another~~ an  
16 association of a second executable feature associated with the ~~other~~ second  
17 graphics element and a new state of operation of the software application;

18 storing the information descriptive of the new state of operation of the  
19 software application and the ~~another~~ association between the second executable  
20 feature and the second graphics element in the map data structure[[:]]; ~~the other~~  
21 association associating the other executable feature with the other graphics element;

22 selecting from the map data structure ~~the other~~ an executable feature that has  
23 not been previously executed, according to a sequence determined by one of a  
24 plurality of deterministic modes of execution of the software application; and  
25

1           executing the selected ~~other~~ executable feature.

2  
3   24.   **(Currently Amended)** The method of claim 22, wherein the retrieving  
4   comprises capturing information pertaining to the second graphics element.

5  
6   25.   **(Currently Amended)** The method of claim 22, wherein the storing  
7   comprises updating an indicator associated with the second graphics element when  
8   an executable feature stored in association with the graphics element is executed.

9  
10   26.   **(Currently Amended)** The method of claim 22, wherein the selecting  
11   comprises reviewing an indicator to determine an executable feature not previously  
12   executed.

13  
14   27.   **(Currently Amended)** The method of claim 22, wherein one of the plurality  
15   of deterministic modes of execution of the software application ~~the selecting~~  
16   ~~comprises selecting executable features in~~ includes a depth-first mode of operation.

17  
18   28.   **(Currently Amended)** The method of claim 22, wherein one of the plurality  
19   of deterministic modes of execution of the software application ~~the selecting~~  
20   ~~comprises selecting executable features in~~ includes a breadth-first mode of  
21   operation.

22  
23   29-39. **(Canceled)**

24  
25   40.   **(New)** The method of claim 1, wherein a state of operation of the software



1 application includes a distinctive set of graphics elements, content, and associated  
2 actions of the software application during execution.  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25